

1. Spring Boot Basics

1. What is Spring Boot, and how is it different from the Spring Framework?
 2. What are the advantages of using Spring Boot?
 3. What is the purpose of `@SpringBootApplication` annotation?
 4. How does Spring Boot provide embedded servers? Which servers does it support?
 5. What are **Spring Boot starters**, and why are they useful?
 6. What is the role of Spring Boot Initializer?
 7. How does Spring Boot reduce boilerplate configuration?
 8. What is the difference between `application.properties` and `application.yml`?
 9. How do you create a simple Spring Boot application from scratch?
 10. Explain about the bean scopes.
-

2. Annotations

11. What is **Inversion of Control (IoC)** in Spring?
 12. What is **Dependency Injection (DI)**, and how does it work in Spring Boot?
 13. What are the different types of Dependency Injection in Spring Boot?
 14. Difference between `@Inject`, `@Autowired`, and `@Resource`?
 15. What is the role of `@Autowired`, and how does it resolve dependencies?
 16. What is the difference between `@Component`, `@Service`, and `@Repository`?
 17. How do you manually define a bean using `@Bean` annotation?
 18. What is the difference between `@Primary` and `@Qualifier` annotations?
 19. Difference between constructor-based DI compare to setter-based DI.
 20. How can you inject values into a bean using `@Value` annotation?
-

3. Spring Boot REST API Development

21. How do you create a REST API in Spring Boot?
22. What is `@RestController`, and how is it different from `@Controller`?
23. What is `@RequestBody`, and how does it work?

- 24. What is `@PathVariable`, and how is it different from `@RequestParam`?
 - 25. How do you handle exceptions in Spring Boot? (`@ExceptionHandler`, `@ControllerAdvice`)
 - 26. How do you enable **CORS (Cross-Origin Resource Sharing)** in Spring Boot?
 - 27. What is `ResponseEntity`, and when should you use it?
 - 28. How do you implement request validation using `@Valid` and `@Validated`?
 - 29. How do you document APIs using Swagger/OpenAPI?
 - 30. How can you limit the rate of API requests in Spring Boot?
-

4. Spring Boot Database and JPA

- 31. What is Spring Data JPA, and how does it simplify database operations?
 - 32. How do you configure a database connection in Spring Boot?
 - 33. How do you use `@Entity`, `@Table`, `@Column` in JPA?
 - 34. What is the difference between `CrudRepository`, `JpaRepository`, and `PagingAndSortingRepository`?
 - 35. How does **lazy loading** and **eager loading** work in JPA?
 - 36. What are the different types of relationships in JPA (`@OneToOne`, `@OneToMany`, `@ManyToMany`)?
 - 37. What is the **N+1 problem** in JPA, and how can you fix it?
 - 38. How do you implement database migrations in Spring Boot using Flyway or Liquibase?
 - 39. How does **Spring Boot integrate with Redis** for caching?
 - 40. How do you perform custom queries using `@Query` annotation?
-

5. Spring Boot Security (High-Level)

- 41. What is Spring Security, and why is it used?
- 42. How do you implement **basic authentication** in Spring Boot?
- 43. What is JWT, and how does it work in Spring Security?
- 44. What is OAuth2, and how can you implement it in Spring Boot?

45. How do you define role-based access control (RBAC) in Spring Security?

6. Spring Boot Actuator & Monitoring

46. What is **Spring Boot Actuator**, and why is it useful?

47. How do you enable and configure Actuator endpoints?

48. How can you monitor a Spring Boot application using Actuator?

49. What are the important Actuator endpoints? (/actuator/health, /actuator/metrics, etc.)

50. How do you secure Actuator endpoints in a Spring Boot application?